

Inférer et simuler un modèle biologique décrivant l'adhérence entre cellules

Fabien Corblin^{1,2}, Eric Fanchon¹, Laurent Trilling²

¹ LCM et LIM, Institut de Biologie Structurale Jean Pierre Ebel,
CEA-CNRS-Université Joseph Fourier,
41, rue Jules Horowitz, 38027 Grenoble Cedex 1, France,

² IMAG-LSR, Université Joseph Fourier BP 53, 38041 Grenoble Cedex 9, France

laurent.trilling@imag.fr

Résumé

L'adhésion entre cellules joue un rôle critique dans la formation des tissus et des organes. Elle intervient aussi dans le contrôle des leucocytes traversant l'endothélium des vaisseaux sanguins. Notre connaissance de ce phénomène est actuellement partielle. Même si certaines protéines impliquées et leurs interactions sont identifiées, d'une part d'autres intervenants restent certainement à découvrir et d'autre part les valeurs des paramètres cinétiques doivent être déterminées. De tels problèmes rendent nécessaires des modélisations qui œuvrent au niveau qualitatif et qui fournissent au biologiste une large palette de fonctionnalités, allant de l'inférence de modèles à partir de données comportementales, à la simulation, en passant par la vérification de propriétés et la proposition d'expériences significatives.

La Programmation Logique avec Contraintes (PLC) apparaît a priori comme un très bon candidat face à cette problématique dans la mesure où elle propose une seule spécification pour plusieurs besoins. Nous présentons la définition et la mise en oeuvre en PLC d'un type bien identifié de modèles : les réseaux logiques asynchrones multivalués dus à R. Thomas, en soulignant l'intérêt de la composition de contraintes (booléennes et numériques). Nous illustrons les capacités multi-fonctionnelles de cette approche en étudiant un sous-système relatif à l'adhésion cellulaire représenté comme un réseau logique. Nous nous intéressons particulièrement à la présence d'états stationnaires et à celle de comportements réparateurs de l'adhérence après une perturbation.

1 Introduction

Avec le développement de projets systématiques de type génomique, la quantité de données moléculaires explose. Il est à présent clair que la biologie est entrée dans une nouvelle ère dans laquelle tous les composants moléculaires doivent être assemblés dans un *système* pour atteindre de nouveaux niveaux de compréhension.

Jusqu'à présent les biologistes synthétisent les résultats sous forme de graphes informels accompagnés de commentaires en langage naturel. La cellule étant un système complexe, son comportement résulte de la dynamique de réseaux moléculaires constitués de boucles de rétroaction (régulation génétique, régulation allostérique) et dont les composants interagissent de façon hautement non-linéaire (présence de sigmoïdes). A cette complexité intrinsèque s'ajoute le fait que nos connaissances sur un système particulier sont encore en général très lacunaires et qualitatives. Pour ces raisons, nous pensons qu'il existe avec le développement de la biologie systémique un fort besoin pour de nouvelles méthodes de modélisation et pour des outils destinés à aider les biologistes à construire des modèles de processus cellulaires, explorer leurs propriétés dynamiques, inférer des paramètres à partir d'observations comportementales, générer de nouvelles hypothèses, et faire des simulations.

La démarche de modélisation employée est la suivante : à partir du graphe de réactions chimiques (phosphorylation, déphosphorylation, complexation, etc.) nous dérivons un système d'équations différen-

tielles ordinaires en nous plaçant dans le cadre de l'approximation quasi-stationnaire. Bien que le formalisme des équations différentielles ne soit pas adapté à un contexte où les paramètres sont totalement inconnus ou seulement partiellement connus, nous voulons conserver le lien avec ce formalisme de façon à pouvoir construire une modélisation plus quantitative lorsque la quantité de données le permettra. Lorsque le système peut se ramener à un système d'Equations Différentielles Linéaires par Morceaux (EDLM) la procédure d'abstraction développée initialement par R. Thomas et E. H. Snoussi [1993] conduit à un formalisme de réseau discret paramétré permettant d'analyser qualitativement la dynamique de réseau. Les variables de concentration ainsi que les paramètres du modèle deviennent des quantités discrètes (booléennes ou entières). Dans le même ordre d'idées, un formalisme discret proche des réseaux de neurones mais également appliqué aux réseaux génétiques a été proposé par Demongeot [5].

Notre objectif principal dans ce papier est de montrer, comme suggéré dans [2], qu'une description formelle d'un système biologique telle que celle de R. Thomas peut être facilement exploitée à l'aide d'une implémentation en Programmation Logique par Contraintes (PLC). Les avantages d'une approche PLC sont (i) que l'implémentation est très proche d'une spécification formelle, ceci permettant de garantir la correction de l'implémentation, et (ii) que différentes requêtes peuvent facilement être formulées sur cette spécification grâce à sa forme logique. La simulation et l'inférence de paramètres en sont des cas particuliers.

Un autre avantage dans notre contexte est qu'une telle implémentation permet d'exploiter de manière efficace des connaissances partielles. Chaque observation nouvelle peut être traduite sous forme de contrainte et exploitée de façon à réduire l'espace des solutions. D'autre part, même si tous les paramètres du modèle ne sont pas tous instanciés, on veut pouvoir poser des questions concernant l'éventail des comportements possibles (par exemple le nombre maximum d'états stationnaires, l'existence de cycles). Pour réaliser ces tâches il faut tout exprimer en termes de formules logiques : la définition des successeurs d'un état, les résultats expérimentaux ainsi que les requêtes.

Dans notre contexte, les observations sont exprimées sous formes de contraintes qui définissent un ensemble de solutions (une solution est un modèle dont tous les paramètres ont été instanciés). Ceci permet donc de décrire un état de connaissance partiel (c'est pourquoi nous parlons d'ensemble de solutions ou de modèle paramétré). N'est inclus que ce qui est pleinement justifié par les données. D'autre part, le pro-

cessus de modélisation peut commencer sans attendre d'avoir déterminé tous les paramètres. Et surtout le modèle paramétré peut être activement exploité, par exemple pour générer des hypothèses et concevoir des scénarios expérimentaux. Des propriétés générales du réseau, comme le nombre maximum d'états stationnaires, peuvent éventuellement être étudiées. La démarche est intrinsèquement itérative puisque chaque nouvelle observation permet d'ajouter une contrainte et ainsi réduire l'espace des solutions.

Ces principes sont illustrés par la modélisation du phénomène d'adhésion entre cellules endothéliales. Une fois que nous possédons la formalisation logique du modèle discret, nous explorons les propriétés du système, en particulier, nous essayons d'inférer les valeurs des paramètres à partir de comportements observés. Chaque comportement est une contrainte qui s'ajoute aux contraintes courantes.

L'organisation du papier est la suivante : après une courte description du système biologique, nous décrivons le réseau de réactions et ses composants. Dans la section 3 les choix de modélisation et les approximations sont brièvement rappelés. Cela nous permet d'obtenir un système d'EDLM de dimension réduite. La théorie de Thomas-Snoussi [14] est introduite et un modèle est formulé dans ce formalisme discret. Nous expliquons comment le modèle est implémenté en PLC (Prolog IV [3]) et donnons des exemples de requêtes. Nous présentons ensuite succinctement une extension du formalisme aux états dits singuliers basée sur l'approche de Filippov introduite dans ce contexte par de Jong et *al.* [12]. Nous donnons aussi des exemples de requêtes dans ce formalisme étendu. Pour finir une discussion sur les caractéristiques de la mise en œuvre nous amène à présenter les difficultés de modélisation rencontrées et les voies de recherche intéressantes.

2 Description du système biologique

Les parois des vaisseaux sanguins sont constituées de monocouches de cellules endothéliales qui forment une barrière entre le sang et les tissus. Cette paroi, appelée endothélium, joue un rôle central dans la régulation du recrutement des leucocytes (globules blancs) au niveau des blessures et des inflammations. Ce contrôle s'effectue en détectant les changements de composition dans le voisinage des sites infectieux, ce qui déclenche l'expression ou la libération à la surface de la cellule d'une variété de médiateurs.

Les jonctions entre les cellules endothéliales (jonctions adhérentes) résultent de l'interaction entre les protéines de VE-cadhérine (*Vascular Endothelium Cadherin*) qui sont spécialement exprimées au niveau des jonctions. Les VE-cadhérines sont directement res-

ponsables de la maintenance de la perméabilité de l'endothélium [10] et du contrôle du trafic des leucocytes du sang vers les sites d'inflammations [11]. Un contact rapproché entre des leucocytes et les cellules endothéliales entraîne la sécrétion de protéases par les leucocytes, et ainsi un clivage local des VE-cadhérines. Ceci conduit à une cassure des jonctions adhérentes, et permet la migration des leucocytes à travers l'endothélium. Pour éviter une accumulation de ces derniers dans les tissus inflammés, l'endothélium doit être restauré.

Nous nous intéressons dans ce papier essentiellement à ce phénomène de reconstruction de l'endothélium. L'équipe de D. Gulino à l'Institut de Biologie Structurale (IBS) est capable de produire en culture des cellules endothéliales extraites de cordons ombilicaux humains. Dans ces cultures les cellules se multiplient jusqu'à confluence (formation d'un tapis cellulaire jointif). Le passage de leucocytes à travers l'endothélium vasculaire est simulé sur une culture cellulaire en ajoutant des anticorps anti-VE-cadhérines qui déstabilisent les jonctions. Après cette perturbation des trous au niveau des contacts entre cellules sont observables. Après quelques heures les jonctions sont réparées. c'est ce type d'observation que l'on va exploiter.

La structure biochimique du système est informellement illustrée dans la Figure 1. Les VE-cadhérines (variable y sur la figure) sont des protéines membranaires avec une queue dans le cytoplasme et de multiples domaines extracellulaires. La β -caténine est un partenaire de la VE-cadhérine qui se lie à la partie cytoplasmique de celle-ci. La Fig. 1 montre son rôle central dans l'adhésion. Lorsque la concentration x de β -caténine cytoplasmique augmente, celle-ci peut entrer dans le noyau où elle se complexe avec des membres de la famille Tcf (variable w) qui sont des facteurs de transcription contrôlant l'expression de certains gènes. Il est probable que ce déplacement dans le noyau est un processus de transport actif (par opposition à la diffusion passive). Le mécanisme réel est mal caractérisé et nous considérerons deux mécanismes pour le modéliser.

La β -caténine cytoplasmique est dégradée par un complexe appelé protéasome. La dégradation protéasomale est elle-même un processus complexe qui fait intervenir différentes protéines. Nous considérons que ce processus est un module séparé avec une simple entrée, et nous ne le décrirons pas ici.

3 Choix de modélisation

La première étape du processus de modélisation est d'extraire un système d'équations différentielles de faible dimension. Ceci a déjà été décrit [7], aussi nous

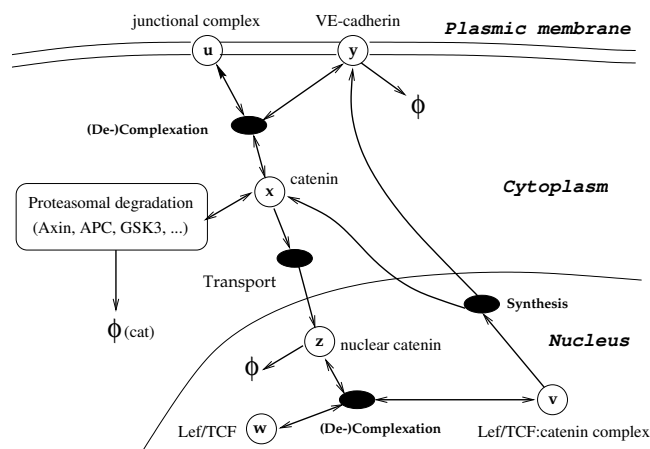


FIG. 1 – Représentation informelle et schématique du système d'adhésion des cellules endothéliales. Ce graphe montre l'architecture générale du système. Les cercles représentent les espèces chimiques et les ovales noirs les réactions chimiques. ϕ représente un processus de dégradation. Les molécules sont distribuées sur trois lieux cellulaires : le cytoplasme, le noyau et la membrane plasmique. Les variables de concentration (x, y, z, u, v et w) sont définies par les couples (variable de concentration, définition verbale) comme suit : (x , β -caténine cytoplasmique non-phosphorylée), (y , VE-cadhérine dans la membrane), (z , β -caténine dans le noyau cellulaire), (u , complexe de β -caténine et trimère de VE-cadhérine), (v , complexe de Lef/Tcf et β -caténine), (w , facteur de transcription Lef/Tcf). Les flèches simples et doubles dénotent respectivement des réactions irréversibles et réversibles.

ne présentons ici que les principes. A partir du graphe des réactions chimiques il est possible de dériver un système d'équations différentielles donnant les taux de variations de la concentration de chaque espèce chimiques. Plus précisément chaque équation est le bilan des processus qui produisent une espèce donnée, et des processus qui conduisent à faire diminuer la concentration de cette même espèce.

Par exemple, considérons la concentration z de la β -caténine dans le noyau. Le taux associé à cette espèce (cf Fig. 1) est informellement donné par :

$$\dot{z} = [\text{transport}] + [\text{decomplexation de v}] \\ - [\text{complexation de w et z}] - [\text{degradation de z}]$$

où le processus de transport est compté parmi les termes de production puisque la β -caténine traverse la membrane nucléaire pour aller dans le noyau. Le transport est bien sûr compté comme un terme de consommation pour la β -caténine cytoplasmique (variable x).

Une simplification majeure est obtenue en considérant que les processus du système agissent à différentes échelles de temps. Le transport, l'expression de gènes et la dégradation sont des processus lents comparés aux réactions de complexation/décomplexation. Ces dernières sont donc en état quasi-stationnaire à tout instant, ce qui signifie que les concentrations des espèces de ces réactions ont une dérivée nulle. Autrement dit certaines concentrations sont liées par une relation algébrique. Un sous-système de deux variables x et z peut alors être isolé. Les quatre autres variables peuvent ensuite être déterminées quand une solution du sous-système (x, z) est connue.

Le sous-système à deux dimensions est le suivant :

$$\begin{cases} \dot{x} = k_{x0} + k_{x1} \cdot \sigma^+(z, s_{z1}) - k_t \cdot f_t(x) \\ \quad - (m_{x1} + m_{x2} \cdot \sigma^+(x, s_{x2})) \cdot x \\ \dot{z} = k_t \cdot f_t(x) - m_z \cdot z \end{cases} \quad (1)$$

Où $\sigma^+(x, s)$ représente une fonction sigmoïde croissante de seuil s , les paramètres k représentent les constantes cinétiques de production, et les paramètres m ceux de dégradation. Notez que dans l'équation pour x le terme de dégradation n'est pas une simple constante m_x multipliée par x , mais fait intervenir un terme sigmoïdal pour prendre en compte le fait que la dégradation des β -caténines dans le cytoplasme n'est pas spontanée mais catalysée par le protéasome. Notez aussi que le terme de transport $k_t \cdot f_t(x)$ apparaît deux fois, comme expliqué précédemment. Nous avons considéré deux manières de représenter *qualitativement* le transport mais nous n'en présentons qu'une dans ce papier (transport dit 'sigmoïdal') : $f_t(x) = \sigma^+(x, s_{x1})$.

4 Théorie de Thomas-Snoussi

Glass [8, 9] et Thomas [16] ont développé une description logique de réseaux de régulation génétique. Elle permet d'analyser *qualitativement* la dynamique de tels réseaux. Comme nous l'avons déjà dit les équations logiques sont dérivées de systèmes différentiels linéaires par morceaux de la forme suivante :

$$\dot{x}_j = h_j(x_1, \dots, x_i, \dots) - m_j \cdot x_j \quad (2)$$

où h_j est une *somme* de produits de fonctions "marche d'escalier" (positives et négatives) dépendantes des variables x_i (pouvant inclure x_j elle-même). Les variables x_i peuvent être vues comme les concentrations des protéines produites par les gènes.

La différence notable entre ce type d'équations et le nôtre (Eq. 1) est que dans notre cas le coefficient dans le terme de dégradation de x n'est pas constant mais contient une fonction $\sigma^+(x, s)$. Le traitement de

Thomas-Snoussi peut néanmoins être généralisé pour traiter des systèmes de ce type.

Un état \mathbf{x} du système est défini par un vecteur (x_1, \dots, x_j, \dots) et un chemin est une séquence d'états.

L'espace des états (des concentrations) est divisé en domaines rectangulaires par l'existence de seuils le long de chaque axe x_i : dans chaque domaine D_k le système différentiel est linéaire, et un point focal F_k peut être associé à D_k : $F_k = \phi(D_k)$ (F_k est le point vers lequel le système *tend* lorsqu'il se trouve dans le domaine D_k). Un autre aspect important est que les paramètres cinétiques sont, comme nous l'avons déjà signalé, faiblement connus ou même complètement inconnus. Snoussi, Thomas et leurs collègues [14, 15, 16] ont développé une méthode dans laquelle ces paramètres sont discrétisés sur l'échelle définie par les seuils. Cela signifie que la position des points focaux dans l'espace continu n'est pas précisément connu, seul est connue le domaine auquel appartient le point focal F_k . Ainsi, les transitions entre domaines peuvent être établies et le *graphe de transitions* construit (les nœuds du graphe étant les domaines).

Après discrétisation, un état \mathbf{X} est associé à chaque état \mathbf{x} : $\mathbf{X} = f(\mathbf{x})$. L'état \mathbf{X} est l'état vers lequel le système *tend* quand il est dans l'état \mathbf{x} . Si une seule variable diffère entre \mathbf{X} et \mathbf{x} , cette variable change de valeur lors de la transition $t \rightarrow t + 1$. Si n variables diffèrent entre \mathbf{X} et \mathbf{x} , n transitions ayant \mathbf{X} comme origine sont considérées car *seulement une* variable peut changer de valeur lors d'une transition. Autrement dit, dans le cadre de cette abstraction, il est naturel de considérer que le réseau est *asynchrone*. Il en résulte, puisque plusieurs transitions sont possibles à partir d'un état donné, que l'évolution du système est non-déterministe.

5 Modèles discrets

Les points focaux s'obtiennent en écrivant qu'en ces points les dérivées s'annulent. Pour simplifier ces équations nous introduisons les paramètres K suivants :

$$\begin{aligned} K_{x,0} &= \frac{k_{x0}}{m_{x1}}, & K_{x,1} &= \frac{k_{x1}}{m_{x1}}, & K_{x,2} &= \frac{k_t}{m_{x1}}, \\ K_{x,3} &= \frac{k_{x0}}{m_{x1} + m_{x2}}, & K_{x,4} &= \frac{k_{x1}}{m_{x1} + m_{x2}}, & K_{x,5} &= \frac{k_t}{m_{x1} + m_{x2}}, \\ K_z &= \frac{k_t}{m_z}, \end{aligned}$$

Par ailleurs, une nouvelle approximation est faite, consistant à remplacer les sigmoïdes (σ) par des fonctions marche d'escalier (\mathfrak{s}). L'équation donnant la composante X du point focal associé à (x, z) devient :

$$X = \frac{k_{x0} + k_{x1} \cdot \mathfrak{s}^+(z, s_{z1}) - k_t \cdot \mathfrak{s}^+(x, s_{x1})}{m_{x1} + m_{x2} \cdot \mathfrak{s}^+(x, s_{x2})}$$

On veut éliminer le dénominateur et se ramener à une *somme* de termes, comme expliqué dans la section précédente. Pour cela on utilise ensuite le fait que $-\mathfrak{s}^+ = \mathfrak{s}^- - 1$ et on regroupe les deux cas $x < s_{x2}$ et $x > s_{x2}$ en une seule équation pour X :

$$\begin{cases} X = \mathfrak{s}^-(x, s_{x2}) \cdot [(K_{x,0} - K_{x,2}) \\ \quad + K_{x,1} \cdot \mathfrak{s}^+(z, s_{z1}) + K_{x,2} \cdot \mathfrak{s}^-(x, s_{x1})] \\ \quad + \mathfrak{s}^+(x, s_{x2}) \cdot [(K_{x,3} - K_{x,5}) \\ \quad + K_{x,4} \cdot \mathfrak{s}^+(z, s_{z1}) + K_{x,5} \cdot \mathfrak{s}^-(x, s_{x1})] \\ Z = K_z \cdot \mathfrak{s}^+(x, s_{x1}) \end{cases}$$

Suivant Thomas et *al.* [2001], nous définissons des opérateurs de discrétisation. Si par exemple, une variable réelle a possède deux seuils ($s_{\text{Sup}}, s_{\text{Inf}}$ avec $s_{\text{Sup}} > s_{\text{Inf}}$) elle est abstraite en une variable discrète $a = d(a)$ comme suit :

$$\begin{aligned} a = 0 &\Leftrightarrow a < s_{\text{Inf}} \\ a = 1 &\Leftrightarrow s_{\text{Inf}} < a < s_{\text{Sup}} \\ a = 2 &\Leftrightarrow s_{\text{Sup}} < a \end{aligned}$$

Nous introduisons les variables discrètes suivantes :

$$X = d_x(X), Z = d_z(Z), x = d_x(x), z = d_z(z).$$

Quatre seuils sont présents dans ce modèle : $s_{x1}, s_{x2}, s_{z1}, s_{z2}$ et par conséquent quatre paires d'ordre de seuils doivent être examinées : $s_{x1} < s_{x2}$ et $s_{z1} < s_{z2}$ (modèle B1), $s_{x1} < s_{x2}$ et $s_{z2} < s_{z1}$ (modèle B2), $s_{x2} < s_{x1}$ et $s_{z1} < s_{z2}$ (modèle B3), $s_{x2} < s_{x1}$ et $s_{z2} < s_{z1}$ (modèle B4). Le seuil s_{z2} n'apparaît pas dans les équations ci-dessus (sous-système à deux variables), mais on le prend en compte car il est nécessaire pour le système complet. Les tables de transitions des modèles B1, B2, B3 et B4 sont données dans Tab. 1, avec les définitions de paramètres suivantes : $K_{a,i} = d_a(K_{a,i})$, $K_{a,i+j} = d_a(K_{a,i} + K_{a,j})$, $K_{a,i-l} = d_a(K_{a,i} - K_{a,l})$, $K_{a,i+j-l} = d_a(K_{a,i} + K_{a,j} - K_{a,l})$.

Il y a neuf paramètres discrets K pour ces quatre modèles, qui sont contraints par leurs définitions : $K_{x,0-2} \leq K_{x,0}$, $K_{x,0} \leq K_{x,0+1}$, $K_{x,0-2} \leq K_{x,0+1-2}$, $K_{x,0+1-2} \leq K_{x,0+1}$, $K_{x,3-5} \leq K_{x,3}$, $K_{x,3} \leq K_{x,3+4}$, $K_{x,3-5} \leq K_{x,3+4-5}$, $K_{x,3+4-5} \leq K_{x,3+4}$, $K_{x,3} \leq K_{x,0}$, $K_{x,3+4} \leq K_{x,0+1}$, $K_{x,3-5} \leq K_{x,0-2}$, $K_{x,3+4-5} \leq K_{x,0+1-2}$, $K_z = 2$. La dernière contrainte provient d'une hypothèse biologique. Nous verrons dans la section 7.2 comment la prouver.

6 Mise en œuvre du modèle en PLC

Nous rappelons brièvement les objectifs généraux d'une implémentation d'un modèle biologique en PLC pour montrer ensuite la concision de la mise en œuvre des modèles de Thomas en PLC et présenter quelques requêtes significatives à la fois biologiquement et du point de vue des performances .

TAB. 1 – Tables de transition pour les modèles B1, B2 (en haut), et B3, B4 (en bas)

x	z	X	Z	X	Z
0	0	$K_{x,0}$	0	$K_{x,0}$	0
0	1	$K_{x,0+1}$	0	$K_{x,0}$	0
0	2	$K_{x,0+1}$	0	$K_{x,0+1}$	0
1	0	$K_{x,0-2}$	K_z	$K_{x,0-2}$	K_z
1	1	$K_{x,0+1-2}$	K_z	$K_{x,0-2}$	K_z
1	2	$K_{x,0+1-2}$	K_z	$K_{x,0+1-2}$	K_z
2	0	$K_{x,3-5}$	K_z	$K_{x,3-5}$	K_z
2	1	$K_{x,3+4-5}$	K_z	$K_{x,3-5}$	K_z
2	2	$K_{x,3+4-5}$	K_z	$K_{x,3+4-5}$	K_z
x	z	X	Z	X	Z
0	0	$K_{x,0}$	0	$K_{x,0}$	0
0	1	$K_{x,0+1}$	0	$K_{x,0}$	0
0	2	$K_{x,0+1}$	0	$K_{x,0+1}$	0
1	0	$K_{x,3}$	0	$K_{x,3}$	0
1	1	$K_{x,3+4}$	0	$K_{x,3}$	0
1	2	$K_{x,3+4}$	0	$K_{x,3+4}$	0
2	0	$K_{x,3-5}$	K_z	$K_{x,3-5}$	K_z
2	1	$K_{x,3+4-5}$	K_z	$K_{x,3-5}$	K_z
2	2	$K_{x,3+4-5}$	K_z	$K_{x,3+4-5}$	K_z

6.1 Objectif

La construction d'un programme en PLC procède classiquement en deux étapes :

1. Spécification logique : dans notre cas, il s'agit d'une part de définir le type de modèle considéré (les réseaux logiques multivalués asynchrones) et d'autre part le modèle biologique étudié (adhérence entre cellules). Ces définitions sont établies à l'aide de prédicats permettant de poser les contraintes auxquelles doivent satisfaire les comportements admis par le modèle biologique.
2. Requêtes : certains paramètres peuvent être très contraints et d'autres beaucoup moins. Dans notre cas, les paramètres sont typiquement les paramètres cinétiques (discrétisés) et les comportements sous la forme de séquences d'états (chemins).

Notre objectif est de montrer comment il est possible, en utilisant la même spécification logique, de poser des requêtes de simulation qualitative (les paramètres cinétiques sont connus, les chemins compatibles sont recherchés), d'inférence (des chemins sont connus, les paramètres cinétiques correspondants sont recherchés) ou mixtes (chemins et paramètres partiellement connus).

Pour cela, on peut rappeler les aspects méthodologiques suivants :

- les contraintes que nous utilisons, sur intervalles entiers avec un nombre fini de valeurs, sont

telles que le résolveur ne fournit qu’une cohérence faible : les réponses sûres ne sont que les négatives (incohérence du système de contraintes) ou les positives associées à une solution instanciée (obtenue par énumération). L’enjeu consiste à déterminer les meilleures heuristiques d’énumération pour un type de requête donnée.

- la spécification logique doit être unique et aussi, dans notre cas, décrire des comportements non déterministes. Ceci implique une utilisation intensive de compositions de contraintes booléennes et numériques (contraintes dites *réifiées*).
- la terminaison des requêtes doit être assurée. Elle l’est théoriquement dans notre cas dans la mesure où les chemins non cycliques ont une taille maximum. Pour des raisons de performance, cette garantie théorique ne suffit évidemment pas et il faut se préoccuper en particulier de minimiser la taille des chemins présents dans une requête.
- enfin, un des moyens importants pour améliorer l’efficacité de la résolution de contraintes sur intervalles consiste à introduire des contraintes redondantes. Ces contraintes, produites à la main, peuvent contribuer considérablement à une meilleure propagation.

6.2 Organisation du programme

Le prédicat `model_asynch_multival(Modele, Chemin)` définissant les réseaux de Thomas est vrai si `Chemin` est un chemin possible du modèle `Modele`. Un chemin est une liste d’états. Un modèle (B1, B2, B3 ou B4) est défini par l’ordre sur les seuils. Il est représenté par une table de transition (voir Tab. 1) et par un ensemble d’inégalités entre ces paramètres.

Exemple : supposons que le modèle `Modele` est connu, défini par une table `T` et les paramètres `P`. Une question typique concernant l’existence d’un état stationnaire est la suivante :

```
Chemin = [S, S], Modele = [T, P],
model_asynch_multival(Modele, Chemin).
```

Ce prédicat exprime simplement que chaque état du chemin `Chemin` doit être un successeur de l’état précédent au sens des réseaux de Thomas et du modèle `Modele`. Sa définition est la suivante :

```
model_asynch_multival(Modele, [Etat]).
model_asynch_multival(Modele, [[Ei, Es]|SChemin]):-
    successeur(Modele, Ei, Es),
    model_asynch_multival(Modele, [Es|SChemin]).
```

Pour définir le prédicat `successeur`, les deux prédicats suivants sont utilisés :

1. `etat_focal(Modele, Ei, Ef)` vrai si `Ef` est l’état focal de l’état `Ei` selon la table `T` du modèle

Modele. Dans une première approche (voir section 8), on peut considérer les contraintes à satisfaire comme une conjonction d’implications traduisant la table `T`.

2. `au_plus_un_saut(B, Ei, Ef, Es)` vrai si (i) `Es` ne diffère de `Ei` que d’une composante au maximum, (ii) la différence entre la valeur de cette composante dans `Es` et la valeur de cette même composante dans `Ei` vaut ± 1 (voir section 4) et (iii) `B` est équivalent à : `Ei` et `Es` sont différents. Les contraintes à satisfaire concernent essentiellement le nombre d’occurrences des composantes différentes dans `Ei` et `Es` (plus petite ou égale à 1).

La définition de `successeur` en découle :

```
successeur(Modele, Ei, Es) :-
    B = bdif(Ei, Ef),
    etat_focal(Modele, Ei, Ef),
    au_plus_un_saut(B, Ei, Ef, Es).
```

On remarque que les contraintes introduites sur `B`, exprimant que `Ei` et `Es` différents est équivalent à `Ei` et `Ef` différents, sont typiques de contraintes redondantes : elles sont déduites de la définition des réseaux de Thomas.

6.3 Exemples de requêtes et résultats

Dans la mesure où les requêtes de simulation ne posent pas de problèmes, nous présentons ici deux types de requêtes intéressants d’un point de vue biologique, portant sur l’existence d’états stationnaires et de réparations après une perturbation.

6.3.1 Etats stationnaires

La requête suivante :

```
constr_Model([b ,1], Modele),
model_asynch_multival(Modele, [E1,E1]).
```

qui vise à trouver les états stationnaire de `B1`, répond `E1 ~ [cc(0,2), cc(0,2)]`, où `cc(0,2)` représente l’intervalle fermé 0-2. Cela ne fournit pas d’information intéressante étant donné que l’ensemble solution recouvre tout l’espace. Une énumération sur `E1` donne 3 possibilités (`[0,0]`, `[1,2]`, `[2,2]`). Mais cette réponse ne dit pas si ces trois états stationnaires peuvent co-exister. Pour obtenir une réponse générale au sujet de ces coexistences, la requête :

```
enum(N, 1, 4), constr_Model([b ,N], Modele),
dif(E1, E2), dif(E1, E3), dif(E2, E3),
model_asynch_multival(Modele, [E1,E1]),
model_asynch_multival(Modele, [E2,E2]),
model_asynch_multival(Modele, [E3,E3]),
enum_Etat(E1), enum_Etat(E2), enum_Etat(E3).
```

qui échoue, donne une réponse plus informative : aucun modèle de la classe B n'autorise la coexistence de plus de deux états stationnaires. On note qu'aucune énumération sur les paramètres n'a été nécessaire pour obtenir ces résultats.

Par ailleurs, on peut vérifier qu'il existe des modèles admettant deux états stationnaires. Par exemple, pour la classe de modèles B3, il y a 5 modèles : l'un avec les états stationnaires $[0, 0]$ et $[2, 2]$ et les quatre autres avec les états $[1, 0]$ et $[2, 2]$. L'état $[0, 0]$ peut être interprété comme l'état sain de la cellule (cellule adhérente) dans lequel la β -caténine est présente dans le cytoplasme et dans le noyau à un niveau bas. L'état $[2, 2]$ signifie que la concentration de β -caténine dans les deux compartiments est au niveau le plus élevé et correspond à un état pathologique.

6.3.2 Réparations après perturbation

Arrivé à ce point de l'analyse, il est intéressant, pour la classe B3 avec l'état stationnaire $[0, 0]$ lorsqu'il est possible, d'étudier l'existence de comportements permettant la restauration des jonctions après perturbation. Plus précisément, lorsque les jonctions sont déstabilisées par les anticorps, les VE-cadhérines sont détruites et les β -caténines sont libérées dans le cytoplasme. Notre modélisation représente une telle perturbation en positionnant initialement le système dans un état ayant une valeur de la concentration x plus élevée par rapport à la valeur dans l'état stationnaire sain. Nous appelons *chemin réparateur* pour B3 un chemin (i) dont l'état initial est un état perturbé, (ii) contenant au moins un état où $z = 1$ (il a été observé que la concentration de la β -caténine après une perturbation augmente [10]), et (iii) se terminant dans l'état stationnaire sain. La requête à poser doit donc vérifier par exemple s'il existe un chemin commençant par l'état $[1, 0]$, se terminant dans l'état $[0, 0]$ et contenant un état $[X, 1]$ où X est quelconque.

Une réponse négative est obtenue pour B3, sans aucune énumération. Par contre, il existe des modèles de B3 avec l'état stationnaire $[0, 0]$ admettant des chemins réparateurs mais commençant par l'état perturbé $[2, 0]$. Il devient alors intéressant, dans la mesure où nous cherchons à isoler une classe de modèles abstraits, de se demander si pour tous ces modèles tous les chemins commençant par $[2, 0]$ sont réparateurs. Il suffit d'ajouter à la requête une condition exprimant l'existence d'un chemin non réparateur commençant par $[2, 0]$. Malheureusement, la réponse est positive. Pour poursuivre, il faudrait isoler une sous-classe de B3 ayant la propriété recherchée. Ce serait possible en extension (en donnant tous les jeux de paramètres possibles) mais difficile à exploiter pour un biologiste : une catégorisation en intention serait plus

indiquée (voir section 8).

La terminaison des requêtes reste évidemment un point clé. Dans notre cas, elle est assurée sans difficulté, dans la mesure où peu de composants et de seuils interviennent : la taille des chemins sans cycles est limitée à 8. C'est pourquoi ce sont des chemins de cette taille qui sont utilisés dans les requêtes pour déterminer des chemins réparateurs et non réparateurs. On sait de plus que le successeur d'un état stationnaire est cet état lui-même et cela fait l'objet d'une contrainte redondante complétant un chemin si celui-ci atteint un état stationnaire. Dans le cas général, des moyens plus sophistiqués peuvent être nécessaires (voir section 8).

7 Introduction des états singuliers

Nous présentons dans ce qui suit la notion d'état *singulier*. Comme nous ne pouvons détailler ici la modélisation en PLC des réseaux les intégrant, seul un problème de modélisation est abordé. Nous donnons ensuite un scénario de requêtes possible.

7.1 Notion d'état singulier

Comme cela est mentionné dans [16] et [12] les frontières entre les domaines adjacents doivent être considérées pour obtenir une description correcte des chemins possibles. Par exemple, dans le cas où deux transitions entre des domaines D et D' , telles que $D \rightarrow D'$ et $D' \rightarrow D$, existent, une situation de conflit apparaît. En effet, soit un chemin partant du domaine D , dès qu'il atteint D' il est amené à revenir en D puis en D' et ainsi de suite. En fait, le chemin reste sur la frontière et peut "glisser" le long de celle-ci (pour atteindre un état de plus faible dimension). Une telle frontière est appelée un "mur noir". Cet exemple montre que les frontières doivent être introduites dans la description comme des états additionnels (nommés états singuliers dans la terminologie de Thomas). Par conséquent, on doit être capable de calculer des transitions partant ou provenant de tels états. Remarquons que les frontières entre domaines réguliers adjacents et aussi les intersections de ces frontières sont des états singuliers. De Jong et al. [12] ont développé une méthode générale et rigoureuse pour calculer de telles transitions, basée sur le concept de solution de Filippov pour des équations différentielles avec membre de droite discontinu. Remarquons que le successeur d'un état peut différer de cet état sur plus d'un composant. Deux règles principales permettent de déterminer les transitions d'un domaine D vers un domaine D' de plus basse dimension, et d'un domaine D vers un domaine D' de plus haute dimension. Un programme Java qui implémente

ces règles a été réalisé par de Jong et *al.* [12] : il permet de faire des simulations qualitatives.

Dans l'optique de poser des requêtes plus générales que des simulations, nous avons modélisé ces deux règles sous forme de contraintes. Le nombre d'états est beaucoup plus grand puisqu'on prend en compte les états singuliers (dans le modèle B nous avons 25 états dont 9 états réguliers), et les prédicats définissant le successeur d'un état sont plus complexes. En pratique, le fait que les frontières et les intersections de frontières sont des états signifie que les variables discrètes x et z peuvent prendre des valeurs intermédiaires. Pour faire simple, on peut voir ces valeurs logiques intermédiaires comme des demi-entiers. Par exemple en dimension 1, une frontière entre les domaines étiquetés par $x = 1$ et $x = 2$ est associée à $x = 3/2$. En dimension n les intersections de frontières conduisent à des états singuliers de plus petite dimension qui doivent aussi être considérés. En d'autres termes, plusieurs composantes d'un état peuvent être des demi-entiers. Le nombre de valeurs demi-entières dans un état est appelé l'*ordre* de cet état.

La spécification logique en PLC des réseaux possédant de tels états ne peut être détaillée ici. Néanmoins, il est intéressant de mentionner une des difficultés de modélisation rencontrée dans la mesure où elle présente un caractère général. Dans cette extension, les successeurs d'un état singulier S sont définis à partir du plus petit hyperrectangle contenant les états focaux des états réguliers adjacents à S . Il est possible de définir intentionnellement ces états focaux et le problème se pose donc dans les termes suivants : étant donné un système de contraintes, exprimer les contraintes déterminant la solution minimale (et la maximale) de ce système. Dans le cas général, il y a aucun espoir de résoudre ce problème. Mais si le nombre N de solutions est fini, on peut utiliser par exemple une contrainte du type $\min(a, b, c)$ signifiant que a est le minimum de b et c : il faut alors poser de l'ordre de N contraintes de ce type. Dans notre cas, N est le nombre d'états adjacents, et vaut 2^{Ord} où Ord est l'ordre de l'état singulier considéré. Ceci implique que cette méthode ne peut être utilisée que si l'ordre est connu. Deux attitudes sont alors possibles : soit utiliser un processus suspensif en attente de l'ordre pour poser les contraintes de minimisation, soit déterminer a priori pour un système donné l'ordre maximal (qui peut être effectivement faible).

7.2 Exemples de résultats

Les résultats présentés ici concernent des questions d'atteignabilité, d'identification d'états stationnaires, et d'existence de chemins réparateurs. Toutes les requêtes concernent le modèle B3 exclusivement. Nous

notons s_{zi} la valeur discrète demi-entière correspondant au seuil s_{zi} . Pour le modèle B3 nous avons $s_{z1} < s_{z2}$, et donc $s_{z1} = 1/2$ et $s_{z2} = 3/2$.

1. Tentons de montrer que $K_z = 2$ peut être déduit des observations que les chemins de restauration passent par un état où z est au-dessus du seuil s_{z2} . Comme précédemment, nous supposons que le système possède un état stationnaire sain (la VE-cadhérine n'est pas synthétisé continuellement, qui signifie que z est au dessous de s_{z2}). Il a été observé que lorsqu'on perturbe le système à partir d'un tel état le système passe par un état où la VE-cadhérine est produite (z au dessus de s_{z2}). Il est évident que cela n'est possible que s'il existe une transition d'un état où $z = s_{z2}$ à un état où $z > s_{z2}$. L'affirmation de l'existence de cette transition et la contrainte $K_z \neq 2$ conduit à un échec avec l'aide d'une énumération sur les états de la transition (aucune énumération sur les paramètres n'est nécessaire). Ainsi $K_z = 2$ est démontré, et nous considérons ce paramètre comme instancié dans la suite.
2. Dans les modèles B il y a quatre états singuliers d'ordre 2 (ces états sont réduits à des points étant donné que le système est bidimensionnel). Démontrons qu'aucun état d'ordre 2 n'est atteignable à partir d'un état régulier (ordre 0) pour les valeurs de paramètres suivantes : $K_{x,3} = 0$, $K_{x,3+4} = 1$ et $K_{x,0} = 0$. La requête est exprimée ainsi : les paramètres de B3 sont contraints aux valeurs données précédemment ; un chemin **Chemin** de longueur 3 (la longueur nécessaire pour aller d'un état d'ordre 0 à un état d'ordre 2) est créé ; son premier et son dernier élément sont contraints respectivement à être un état régulier et un état d'ordre 2. De plus **Chemin** est contraint à être un chemin de B3. Un échec est obtenu en énumérant sur **Chemin** (mais pas sur les paramètres). La proposition ci-dessus est ainsi prouvée.
3. Une requête analogue permet de montrer la propriété générale à tous les modèles de B3 qu'aucun état $[x, z]$ avec x entier (*entre* deux seuils) et z demi-entier (*sur* un seuil), n'est un "mur noir".
4. Nous pouvons aussi démontrer qu'il n'existe pas de chemin réparateur si $K_{x,3+4}$, $K_{x,3+4-5}$ et $K_{x,0+1}$ sont tous égaux à 2. Il s'agit du cas où une concentration de la β -caténine dans le noyau provoquerait toujours une augmentation de la concentration de la β -caténine cytoplasmique et donc mènerait à un état stationnaire où la β -caténine serait produite en permanence. Ceci élimine 10 modèles sur 50.

8 Discussion

Nous sommes conscients du fait que le réseau présenté ici, bien qu'utile, reste de petite taille et qu'il doit être étendu pour rendre mieux compte de la réalité biologique. Si les performances en espace et en temps sont acceptables et encourageantes pour ce réseau, le resteront-elles si un plus grand nombre de protéines ou de gènes et de seuils sont introduits ? La réponse à cette question peut être appréciée selon les critères suivants :

- coût en espace, qui dépend du nombre de contraintes attachées à un chemin. Ce nombre est clairement une fonction linéaire du nombre d'états du chemin. Le nombre d'équations exprimant la liaison entre un état et son successeur est, comme nous l'avons vu, une fonction linéaire du nombre de composants d'un état, c'est à dire de la dimension du système. Il reste à considérer le nombre d'équations nécessaires pour définir un état focal. Dans l'implémentation que nous avons décrite et qui utilise des tables de transitions, ce nombre varie exponentiellement avec la dimension. Cependant notre mise en œuvre est moins coûteuse. Elle s'appuie sur le fait que chaque composant d'un état focal d'un état E est défini à partir d'un nombre réduit de composants de cet état E. Ce nombre de composants est associé à la connectivité du graphe d'interaction entre les composants biologiques qui est habituellement faible. Il en ressort que le nombre d'équations définissant un état focal varie, au pire, exponentiellement avec la connectivité du réseau (et non avec sa dimension). Une autre amélioration permet de réduire au mieux l'intervalle d'un composant de l'état focal. Plus précisément, nous avons été amené pour cela à résoudre le problème suivant : soit X_f un composant défini par exemple par $X_f = K1*B1 + K2*B2 + K3*B3 + K4*B4$ où $B1, B2, B3$ et $B4$ sont quatre booléens exclusifs dont au moins l'un est vrai, comment réduire au mieux l'intervalle de X_f suivant la connaissance sur les booléens sans utiliser des représentations avec intervalles disjoints (qui sont coûteuses).

Pour conclure, ce problème de coût en espace ne semble pas crucial. Expérimentalement des chemins de $B3$ contenant 30 états sont facilement traités.

- coût en temps, qui dépend du nombre de contraintes et (principalement) du nombre d'énumérations. Des heuristiques pertinentes doivent être recherchées pour réduire ce coût. En particulier, ce coût est réductible en réduisant le nombre de modèles à considérer, grâce à la construction de classes d'équivalence regroupant des modèles admettant les mêmes graphes de transitions entre états. Une telle classification est

réalisable d'après la définition d'un successeur.

Aussi, des contraintes spéciales peuvent être posées pour des requêtes particulières. Par exemple Devloo [6] utilise les propriétés énoncées dans [14] et la technologie CSP pour découvrir efficacement les états stationnaires de modèles asynchrones complètement instanciés. Cette approche pourrait certainement être utilisée aussi pour induire des paramètres de modèles à partir d'informations concernant les états stationnaires.

Enfin, un autre problème important est le choix des deux langages à fournir aux utilisateurs d'un tel système : celui permettant d'exprimer des propriétés et celui fournissant les résultats.

Dans [1], le langage proposé pour exprimer des propriétés est CTL (*Computational Tree Logic*), en conjonction avec des algorithmes de *model-checking* qui vérifient des propriétés CTL. Cette étude considère que le système biologique est connu. Dans [13], CTL est aussi proposé pour spécifier des propriétés de modèles asynchrones multivalués : cette étude vise aussi à trouver des paramètres à partir de comportements, avec des méthodes de *model-checking* appliquées sur des modèles instanciés. Clairement, CTL et les méthodes de *model-checking* sont à considérer, et tout particulièrement le *model-checking* symbolique comme dans [4]. Adapté à notre problème, ce dernier peut nous aider à gagner en efficacité (spécialement pour des requêtes universelles), mais aussi pour la terminaison (pour éviter des chemins trop longs) et enfin pour imposer des comportements via des formules CTL.

La conception d'un langage pour exprimer les résultats représente un défi réel. Ceci est dû au fait que les biologistes désirent des résultats exprimés sous la forme "ces modèles satisfont ces propriétés", c'est à dire intentionnellement et non extensionnellement, comme ils le sont pour l'instant. Il s'agit d'un problème d'apprentissage automatique qui nécessite d'abord une concertation avec les biologistes pour décider des composants élémentaires des formules d'un tel langage.

9 Conclusion

Nous avons d'abord montré comment représenter le système biologique étudié sous la forme d'EDLM adaptées au formalisme des réseaux asynchrones multivalués de R. Thomas. Puis, nous avons donné les grandes lignes de la mise en œuvre de ces réseaux en PLC pour faire ressortir la simplicité et la rigueur de l'approche. Des exemples de requêtes significatives ont permis d'illustrer la souplesse de cette mise en œuvre. Nous avons ensuite introduit rapidement l'extension de ces réseaux pour prendre en compte les

états singuliers en insistant sur la nécessité d’une telle extension : en effet, à moins qu’il n’y ait aucun “mur noir” une description ne prenant en compte que les états réguliers ne paraît guère acceptable. Et enfin, nous avons discuté de la faisabilité de l’approche en termes de nombre de contraintes et de performances, en faisant état des principales difficultés à surmonter et des voies de recherche possibles. On peut remarquer que ces difficultés, soulevées à propos d’un modèle de réseau biologique, relèvent de problèmes plus généraux : par exemple quand il s’agit de poser des contraintes sur des chemins ou d’obtenir des résultats sous une forme intentionnelle.

Par ailleurs des expériences en protéomique, menées à l’IBS (Institut de Biologie Structurale) sont sur le point d’identifier de nouvelles protéines intervenant dans l’adhérence entre cellules. Ces études étendront le réseau à considérer et introduiront de nouvelles variables et aussi de nouvelles contraintes. Un outil tel que celui présenté sera encore plus nécessaire pour discriminer les modèles non significatifs.

Références

- [1] N. Chabrier and F. Fages. Symbolic Model Checking of Biochemical Networks. Computational Methods in Systems Biology. In *Computational Methods in System Biology 2003*, C. Priami (ed.), LNCS 2602, Springer, 149–162, 2003
- [2] J. Cohen. Classification of approaches used to study cell regulation : Search for a unified view using constraints and machine learning. *ETAI, Machine Intelligence*, 18. Linköping Electronic Articles in *Computer and Information Science ISSN*, 1401-9841, vol. 6 No. 25, 2001
- [3] A. Colmerauer. Prolog – Constraints Inside, *Manuel de Prolog, PROLOGIA*, Case 919, 13288 Marseille cedex 09, France, 1996
- [4] G. Delzanno and A. Podelski. Model Checking in CLP. In *Proceeding of the 5th International Conference TACAS’99*, Springer, LNCS 1579, 223–239, 1999
- [5] J. Demongeot, J. Aracena, F. Thuderoz, T.-P. Baum, O. Cohen. Genetic regulation networks : circuits, regulons and attractors. *C. R. Biologies* 326, 171–188, 2003
- [6] V. Devloo, P. Hansen and M. Labbé. Identification of All Steady States in Large Biological Systems by Logical Analysis. *Bulletin of Mathematical Biology*, 65, 1025–1051, 2003
- [7] E. Fanchon, F. Corblin, L. Trilling, B. Hermant and D. Gulino. Modeling the Molecular Network Controlling Adhesion Between Human Endothelial Cells : Inference and Simulation Using Constraint Logic Programming. In *Computational Methods in Systems Biology 2004*, V. Danos and V. Schachter (Eds.), Lecture Notes in Bioinformatics 3082, 104–118, 2005
- [8] L. Glass, S. A. Kauffman. Co-operative components, spatial localization and oscillatory cellular dynamics. *J. Theor. Biol.*, 34, 219–237, 1972
- [9] L. Glass, S. A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *J. Theor. Biol.*, 39, 103–129, 1973
- [10] D. Gulino, E. Delachanal, E. Concord, Y. Genoux, B. Morand, M. O. Valiron, E. Sulpice, R. Scaife, M. Alemany and T. Vernet. Alteration of Endothelial Cell Monolayer Integrity Triggers Resynthesis of Vascular Endothelium Cadherin. *The Journal of Biological Chemistry*, 273, 29786–29793, 1998
- [11] B. Hermant, S. Bibert, E. Concord, B. Dublet, M. Weidenhaupt, T. Vernet and D. Gulino-Debrac. Identification of Proteases Involved in the Proteolysis of Vascular Endothelium Cadherin during Neutrophil Transmigration. *The Journal of Biological Chemistry*, 278, 14002–14012, 2003
- [12] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative Simulation of Genetic Regulatory Networks Using Piecewise-Linear Models. *Bulletin of Mathematical Biology*, 66, 301–340, 2004
- [13] S. Peres and J. P. Comet. Contribution of Computational Tree Logic to Biological Regulatory Networks : Example from *Pseudomonas Aeruginosa*. In *CMSB 2003*, C. Priami (ed.), LNCS 2602, 47–56, 2003
- [14] E. H. Snoussi and R. Thomas. Logical Identification of All Steady States : The Concept of Feedback Loop Characteristic States. *Bulletin of Mathematical Biology*, 55, 973–991, 1993
- [15] D. Thieffry, M. Colet and R. Thomas. Formalisation of Regulatory Networks : a Logical Method and Its Automation. *Math. Modelling and Sci. Computing*, 2, 144–151, 1993
- [16] R. Thomas and M. Kaufman. Multistationarity, the Basis of Cell Differentiation and Memory. II. Logical Analysis of Regulatory Networks in Term of Feedback Circuits. *Chaos*, 11, 180–195, 2001